```
import numpy as np
import matplotlib.pyplot as plt

def cart2pol(x, y):
    rho = np.sqrt(x**2 + y**2)
    phi = np.arctan2(y, x)
    return(rho, phi)

def pol2cart(rho, phi):
    x = rho * np.cos(phi)
    y = rho * np.sin(phi)
    return(x, y)
⟶
def sum2(x, y):
⟶return tuple(map(sum, zip(x,y)))

def sum3(x, y, z):
⟶return tuple(map(sum, zip(x,y, z)))




def mag(x, y):
⟶return(np.sqrt(x**2+y**2))
def acc(x, y):
⟶return(-.5*x/mag(x,y)**3, -.5*y/mag(x,y)**3)




def mag_array(r):
⟶return(np.sqrt(r[0]**2+r[1]**2))
def acc_array(r):
⟶return(-.5*r/mag_array(r)**3)



def mag_multi(r1,r2):
⟶return(np.sqrt((r2[0]-r1[0])**2+(r2[1]-r1[1])**2))
def acc_multi(r,r1,r2):
⟶return(-.5*(r-r1)/mag_multi(r1,r)**3-.5*(r-r2)/mag_multi(r2,r)**3)



def int_q_array(r,v,dt):
⟶r = r + dt*v
⟶return(r)
def int_v_array(r,r1,r2,v,t):
⟶# print('v before', v)
⟶# print('acc',acc_multi(r,r1,r2))
⟶v = v + t*acc_multi(r,r1,r2)
⟶# print('v after', v)
⟶return(v)


def v_magnetic_calc(r, v, B, dt):
⟶theta = B[2]*dt
⟶B_unit = B/np.linalg.norm(B)
⟶
    return(v+np.sin(theta)*np.cross(B_unit,v)+(1-np.cos(theta))*np.cross(B_unit,np.cross(
    B_unit,v)))


def v_damped(r, v, w_0, dt):
⟶v - w_0**2*dt
⟶return(v)

```

```
66
67
68
69
70
71
72
73
74  def plotting(x, y):
75  ——→fig2, ax5 = plt.subplots()
76  ——→ax5.set_ylabel('(E(t)/E_0-1)/t^4')
77  ——→ax5.set_xlabel('Time/Period')
78  ——→ax5.set_title("Energy Ratio - Forest Ruth")
79  ——→ax5.plot(x,y)
80  ——→ax5.legend(('Runge-Kutta', 'Forest Ruth'), loc='upper right')
81  ——→plt.show()
82
83  def plot_polar(r, theta):
84  ——→fig1, ax3 = plt.subplots()
85  ——→ax3=fig1.add_subplot(111, projection='polar')
86  ——→ax3=fig1.add_subplot(111)
87  ——→ax3.plot(y_val,x_val)
88  ——→ax3.plot(theta_comp, r_comp, 'o', markerfacecolor='none', markeredgecolor='r')
89  ——→ax3.set_rmax(.5)
90  ——→ax3.set_rticks([3, 6, 9, 12])  # less radial ticks
91  ——→ax3.set_rlabel_position(-22.5)  # get radial labels away from plotted line
92  ——→ax3.grid(True)
93  ——→ax3.set_title("Forest Ruth", va='bottom')
94  ——→plt.show()
```